

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra informatiky**

**Počítačová podpora výuky matematické analýzy**  
Computer support of mathematical analysis education

# Prohlášení

Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. 5. 2009

.....

# Poděkování

Na tomto místě bych rád poděkoval vedoucímu bakalářské práce, Mgr. Karlu Klosovi za podporu, cenné názorné rady a odborné konzultace. Dále bych rád poděkoval své sestře Miloslavě Šepsové za cenné informace v oblasti výuky na středních školách.

# Abstrakt

Cílem práce je navrhnout a implementovat software pro podporu výuky matematické analýzy vyučované na středních školách (2. ročník). Výsledný software by se měl skládat z výukové, testovací a názorné části. K softwaru bude dále potřeba vytvořit webovou podporu, kde bude program volně ke stažení. Součástí webu bude i nápověda potřebná k užívání programu.

# Klíčová slova

.NET Framework, Visual Studio, C#, výukový software, matematická analýza, graf

# Abstract

The main object of this work is to design and implement system for support education of mathematical analysis teaches in second school (2. year-class). Resulting software would consist of learning, testing and visual part. Software will need to create a web support where the program would be free to download. One part of web support will be help, necessary to use the program.

# Keywords

.NET Framework, Visual Studio, C#, learning software, mathematical analysis, graph

# Seznam použitých symbolů a značek

C#	C Sharp
CSS	Cascade Style Sheets
DLL	Dynamic link library
HTML	Hyper Text Markup Language
PC	Personal Computer
SP1	Service Pack 1
SQL	Structured Query Language
XML	eXtended Markup Language

# Obsah

<b>1</b>	<b>Úvod.....</b>	<b>9</b>
1.1	Důvody, proč používat počítače ve výuce .....	9
1.2	Funkce programu.....	9
1.2.1	Funkce prezentace učiva .....	9
1.2.2	Funkce řízení učení a vyučování .....	10
1.2.3	Funkce organizační .....	10
1.2.4	Funkce kontrolní.....	10
1.3	Další součásti programu.....	10
1.4	Kritéria programu .....	11
1.4.1	Dokumentace .....	11
1.4.2	Ovládání .....	11
1.4.3	Nápověda.....	11
<b>2</b>	<b>Pedagogické principy .....</b>	<b>13</b>
2.1	Cílevědomost.....	13
2.2	Soustavnost .....	13
2.3	Aktivnost.....	13
2.4	Názornost .....	13
2.5	Uvědomělost .....	14
2.6	Trvalost .....	14
2.7	Přiměřenost .....	14
2.8	Individuálnost.....	15
2.9	Emocionálnost.....	15
2.10	Jednotnost .....	15
<b>3</b>	<b>Etapizace tvorby výukových programů.....</b>	<b>16</b>
3.1	Tvorbu a výrobu výukových programů budeme rozdělovat do etap .....	16
3.1.1	Námět by měl obsahovat tyto údaje .....	16
3.1.2	Projekt výukového programu .....	16
3.1.3	Příprava obsahu výukového programu.....	17
3.1.4	Blokové schéma výukového programu .....	17
3.1.5	Ověření v pedagogické praxi.....	17
<b>4</b>	<b>Požadavky.....</b>	<b>18</b>
4.1	Hardwarové požadavky .....	18
4.2	Softwarové požadavky.....	18
4.2.1	Vývojové požadavky.....	18

4.2.2	Provozní požadavky .....	18
4.3	Požadavky na webovou podporu .....	18
4.4	Požadavky na uživatele .....	18
4.5	Funkční požadavky .....	18
4.5.1	Vlastnosti funkcí .....	18
4.5.2	Typy funkcí .....	19
4.6	Programátorské minimum .....	19
4.7	Programátorské maximum .....	19
4.8	Nefunkční požadavky .....	19
<b>5</b>	<b>Analýza .....</b>	<b>20</b>
5.1	Diagramy .....	20
5.1.1	Třídní diagram .....	20
5.1.2	Use-case diagram .....	21
5.1.3	ER Diagram .....	21
5.1.4	Lineární zápis .....	21
5.1.5	Lineární zápis typů vztahů .....	21
5.2	Datová analýza (datový slovník) .....	22
5.3	Funkční analýza .....	22
5.3.1	Kontextový diagram .....	22
5.3.2	0. Úroveň .....	23
5.3.3	1. Úroveň .....	24
<b>6</b>	<b>Implementace .....</b>	<b>25</b>
6.1	Technologie .....	25
6.1.1	.NET Framework 3.5 SP1 .....	25
6.1.2	C# .....	25
6.1.3	HTML .....	25
6.1.4	CSS .....	26
6.1.5	SQL .....	26
6.1.6	XML .....	26
6.2	Inspirace z jiných programů .....	26
6.2.1	Deskriptivní geometrie 2.02 .....	26
6.2.2	Graph .....	27
6.2.3	Fyzika zajímavě – Optika .....	27
6.3	Předvedení programu .....	27
6.3.1	Přihlašovací vstupní formulář .....	27
6.3.2	Vykreslení grafu .....	28

6.3.3	Výuková část .....	29
6.3.4	Testovací část .....	29
6.3.5	Správa otázek.....	30
6.3.6	Správa uživatelů.....	30
6.4	Předvedení webové podpory softwaru.....	31
<b>7</b>	<b>Závěr.....</b>	<b>32</b>
<b>8</b>	<b>Literatura .....</b>	<b>33</b>



# 1 Úvod

## 1.1 Důvody, proč používat počítače ve výuce

Výukové programy v poslední době představují dynamicky rozvíjející se oblast elektronických didaktických médií. Postupným zaváděním výukových programů do edukačního procesu dochází k dalšímu zkvalitňování výuky, avšak za dodržení nových trendů v edukačním procesu a požadavků kladených na edukátora. Tvorba výukových programů v oblasti technického vzdělávání je velmi náročná, dáno zřejmě množstvím termínů, definic, grafickým zpracováním animací aj., proto některá nejsou v tak kvalitní formě. V informační společnosti nelze žáky pro život připravit pouze pomocí učebnic. Je zapotřebí žáky naučit rychle a efektivně vyhledávat informace z různých informačních zdrojů, analyzovat je, vyčlenit podstatné od vedlejšího, nepodstatného a vhodně je využít. Zvláště generace, které neprošly na základní škole informační výchovou, mají problém plně porozumět datům prezentovaným v elektronické podobě.

Počítač může studenta informovat o výukových cílech, připravovat mu individuální plán studia (např. na základě testu jeho předběžných znalostí a zvolené úrovně aspirací) a harmonogram studia.

Může organizovat jednotlivé studijní činnosti a bezprostředně řídit osvojovací proces s využitím výsledků počítačové kontroly znalostí. Řízení osvojovacího procesu se může uskutečňovat formou výukového dialogu, v němž se uplatní programem zajišťovaná zpětná vazba, spočívající v tom, že na základě typu odpovědi (případně diagnostiky chyby s přihlédnutím k dosavadní historii učení) se programem řídí další činnost s různým stupněm volnosti poskytované studentovi.

Méně dokonalé systémy zpravidla nemají mechanismus vyhodnocování předpokládaného typu odpovědi studenta. Proto rozhodují o dalším kroku buď podle pevného programu (pak je počítač v podstatě jen automatickým obracečem stránek). V prvním případě jde vlastně o osvojovací proces bez zpětné vazby, v druhém se zpětná vazba uzavírá přes studenta, což omezuje její objektivnost. Různí teoretici tyto systémy velmi vyzdvihují s tím, že dávají studentovi více volnosti a také autoři programů se k nim rádi uchylují. Tvorba takových programů je neskonale snadnější než programů se zpětnou vazbou přes program. Jejich výhodou je, že není třeba analyzátoru odpovědi, který u jiných typů než výběrová, numerická a porovnání řetězců, není nijak snadný. Navíc absolutně dokonalý analyzátor a absolutně dokonalé předvídání odpovědi neexistuje a každá chyba v rozpoznání správné odpovědi má fatální dopad na vztah studenta k výukovému programu.

## 1.2 Funkce programu

### 1.2.1 Funkce prezentace učiva

Výukový program je především souborem informací, které různými formami prezentuje. Měl by sdělit studentovi nové informace, dát mu možnost osvojit si nové vědomosti a dovednosti. Tyto informace by měl uvést v co nejširších souvislostech. Úkolem výkladové části výukového programu je zprostředkovat studentovi nové poznatky, a to formou vhodnější než učebnice a někdy třeba i zábavnější než výklad učitele. Jednotlivé významem související informace by měly být navzájem propojeny. Výkladová část ideálního výukového programu by

## 1 ÚVOD

měla uživateli umožňovat volný pohyb ve všech kapitolách i modulech programu, tisknout výklad či vytvářet si vlastní poznámky pomocí kopírování textu přes schránku do jiných aplikací. Tak jako u správné učebnice by součástí výkladové části měl být obsah jednotlivých témat a rejstřík či vyhledávání. Látka obsažená ve výkladové části by samozřejmě měla odpovídat osnovám předmětu, pro který je výukový program určený. Někdy bývá výkladová část výukového programu obohacena kromě audio a video ukázek a obrázků týkajících se probíraného tématu ještě o simulaci probíraných jevů. Ta umožňuje nahradit některé obtížné či vůbec proveditelné pokusy a přiblížit žákům studovaný jev. Správná simulace by měla popisovat daný problém v grafické podobě, dovolovat interaktivně měnit parametry a okamžitě sledovat vliv těchto změn na výsledek. Žák přestává vnímat probíranou tematiku pasivně a staticky, ale může si sám aktivně prozkoumat nejrůznější aspekty studovaných jevů.

### 1.2.2 Funkce řízení učení a vyučování

Výukový program je současně didaktickým prostředkem, který řídí edukační proces, jednak edukanta, ale rovněž i edukátora. Jedním z nejrozšířenějších způsobů využití výpočetní techniky na podporu výuky je procvičování látky. Procvičovací část dobrého výukového programu by uživateli měla umožňovat časově neomezený pobyt v této části, volné přecházení do části výkladové a zpět. Žák by měl mít možnost výběru procvičovaného tématu, stanovení počtu otázek, zpětné zařazování špatně zodpovězených a ukládání výsledku procvičování.

### 1.2.3 Funkce organizační

Výukový program umožňuje žákovi orientaci v obsažených informacích a umožňuje mu snadno organizovat jeho činnost. Důležitou vlastností výukových programů je i to, jakým způsobem působí na své uživatele. Toto z velké části ovlivňuje grafické zpracování programu i případná zvuková kulisa, která doprovází práci s ním. Obecně platí, že čím je nižší věk uživatelů, tím je třeba zvolit větší písmo, více grafických prvků a barev.

Všechny programy by také měly jít okamžitě ukončit. Tam, kde se předpokládá, že se k nim studenti budou vracet, je třeba umět uložit rozpracovaný stav, aby nemuseli začínat vždy od začátku.

Dále můžeme sledovat, zdali program dovoluje modifikovat strukturu a obsah dat, jako je změna nebo přidání textů, obrázků apod. Takovou vlastnost má ovšem pouze minimum výukových programů.

### 1.2.4 Funkce kontrolní

Výukový program by měl změřit stupeň osvojení znalostí a dovedností. To je důležité nejen pro vyhodnocení výsledku učebního procesu, ale především pro jeho řízení (zpětná vazba).

Dobrý výukový program by měl při předání učiva žákovi a kontrole získané úrovně znalostí reagovat i podle výsledků zpětnovazební informace. Přítomnost zpětné vazby má ve výukových programech velký význam, neboť umožňuje počítači reagovat alespoň zčásti jako dobrý pedagog. Ten, když vidí, že žák v nějaké oblasti chybí, předkládá mu doplňující informace, které ho mají vést ke správným závěrům, případně se ve výkladu vrátí k látce, která je žákovi nejasná. Stejně by se měl chovat i dobrý výukový program.

## 1.3 Další součásti programu

Součástí výukových programů bývá i testová část, ve které počítač zjišťuje míru získaných znalostí a dovedností žáka. Tyto testy po vyhodnocení slouží jak pro učitele, tak i pro samotné

## 1 ÚVOD

studenty, kteří s jejich pomocí mohou zjišťovat, do jaké míry si osvojili vykládanou látku. Kontrolní otázky by měly být voleny tak, aby k jejich správnému zodpovězení bylo nutné pochopení probírané látky. Úlohy by měly být různých úrovní složitosti, od nejjednodušších založených na reprodukci získaných poznatků, až po ty, u nichž musí žák sám úlohu analyzovat a vyvodit správné závěry. Žák i učitel by měl mít možnost vybírat témata testu i počty pokládaných otázek, učitelé by potom měli mít možnost tvořit vlastní testy.

První čtyři uvedené funkce jsou důležité z hlediska pedagogiky a didaktiky. Jsou ale ještě další kritéria, která musí daný výukový program splňovat.

### 1.4 Kritéria programu

#### 1.4.1 Dokumentace

Každý dobrý program by měl být vybaven kvalitně provedenou dokumentací. Její nejdůležitější součástí je manuál, který krok za krokem popisuje, jak s programem pracovat. Měl by být dobře srozumitelný, obsahovat popis typických problémů či často se opakujících chyb, kterých se může dopustit ten, kdo ještě nemá s programem velké zkušenosti.

#### 1.4.2 Ovládání

Další vlastností, která určuje kvalitu programu, je jeho ovládání. Zvláště u výukových programů, které ve velké míře používají děti, rodiče a učitelé, kteří ne vždy mají dostatečné znalosti a dovednosti, jak pracovat s počítačem a počítačovými programy, by jejich ovládání mělo být co nejvíce přirozené a intuitivní. Uživatelé by neměli být rozptylováni od učení s počítačem tím, že část svého času a úsilí budou muset věnovat snaze přijít na to, jak a z jakého místa jsou dostupné všechny funkce programu a jak zvládnout principy jeho ovládání.

S ovládáním úzce souvisí způsob rozložení jednotlivých ovládacích prvků na obrazovce neboli uživatelské rozhraní programu a s ním zase struktura programu. Jde o to, jak jsou z kteréhokoli místa dostupné všechny potřebné funkce a jak jsou vzájemně propojeny různé části či moduly programu, pokud je jich více. Moderní výukové programy, mají-li splňovat požadavky na aktivní tvůrčí přístup ze strany studenta, musí umožňovat volný pohyb mezi jednotlivými moduly bez ohledu na to, jakou funkci plní.

#### 1.4.3 Náповěda

V případě, že již dojde k situaci, kdy si uživatel opravdu neví rady buď s ovládáním programu, nebo s dalším postupem, přichází na řadu nápověda. Ta by neměla chybět v žádném kvalitním programu, ve výukovém zvláště. Jejím úkolem je zajistit, aby si s daným problémem dokázal poradit i ten nejhorší žák. Přitom se při práci nesmí nudit ani ten nejlepší. To znamená, že je třeba zajistit, aby se získaná pomoc týkala skutečně pouze toho problému, který je právě řešen (kontextově závislá) a aby byl její informační obsah pokud možno odstupňován (víceúrovňová).

Tyto výukové programy musí zahrnovat také rozpracování klíčových didaktických otázek, jako je např. stanovení cíle výuky, uplatnění pedagogických zákonitostí prostřednictvím didaktických zásad či diagnostiky studentů. Programy, které si kladou za cíl vzdělávat svého uživatele, by se měly řídit zejména pedagogickými principy. Pedagogické principy jsou jedny z nejdůležitějších pravidel a doporučení, kterými se řídí učitel při přípravě vyučovací hodiny i v jejím průběhu. Mají obecnou platnost v celém poli výchovně vzdělávacího působení. V jednotlivých pedagogických situacích se pak adekvátně modifikují ve shodě se sledovanými

## 1 ÚVOD

výchovnými záměry. Jakékoliv výchovně vzdělávací působení je pak ve své koncepci, tak ve své realizaci vázáno těmito principy a jeho úspěšnost závisí na jejich respektování. To vyžaduje při tvorbě takovýchto programů uplatňovat nejen zkušenosti programátorské, ale i pedagogické a didaktické.

## 2 Pedagogické principy

### 2.1 Cílevědomost

Je výchozím principem každé úspěšně výchovně vzdělávací činnosti. Jde o záměrné a cílevědomé působení na jedince či sociální skupinu. Proto je třeba jasně stanovit cíle ve všech stupních hierarchie pedagogického působení. Tyto cíle je samozřejmě nutné dostatečně zdůvodnit, objasnit a ve vztahu ke studentům motivovat.

Tento princip by se při tvorbě výukového programu měl projevit především při výběru obsahu a výukových metod vhodných pro dosažení stanovených cílů. Součástí tohoto výukového programu a každé jeho tematicky samostatné části je uvedení stanovených cílů a motivace k jejich dosažení uživatelem.

### 2.2 Soustavnost

Klíčovým principem je požadavek soustavnosti. Tento princip zajišťuje, aby veškeré výchovné podněty byly uspořádány do zdůvodněného systému, který umožní jejich osvojování v logickém pořádku. Požadavky a působení pedagoga na výchovu žáka musí odpovídat tak, aby činnosti vychovávaného byly co nejsoustavnější.

Má také za úkol vybrat a systematizovat učivo a stanovit jeho normy.

Princip soustavnosti výrazně ovlivní strukturu a členění programu. Jasná struktura by také měla být vyjádřena uživatelským rozhraním. V tomto programu je výběr tématu a jeho částí názorná a intuitivní. Je doplněn o navigační komponentu poskytující celkový přehled položek a umožňující uživateli rychlé dosažení požadované informace.

### 2.3 Aktivnost

Moderní pedagogika považuje princip aktivity za velmi významný. Tento princip vyjadřuje požadavek na samostatnou činnost jedince. Je třeba aktivizovat jeho poznávací, citové a volní procesy. Přitom je nutné se opírat o jeho zájem a stimulovat jej vhodnou motivací. To umožňuje ho soustavně vést k aplikaci a k praktickému využívání získaných vědomostí a dovedností.

Aktivnost každého jednotlivce nepřímo vyplývá již z nasazení výpočetní techniky do procesu výuky. Je ji však nutné podporovat ze strany programu a jeho působení na uživatele. Tento program umožňuje některé výrazy nechat osvětlit prostřednictvím hypertextových odkazů. V nedaleké budoucnosti lze očekávat již obousměrnou interakci mezi programem a uživatelem prostřednictvím mluvené řeči.

### 2.4 Názornost

Tímto principem se rozumí požadavek vycházet ze smyslového nazírání předmětů a jevů skutečnosti a z jejich obrazů, tj. přímý názor. Je třeba se také opírat o dosavadní představy a zkušenosti studenta, tj. nepřímý názor, a systematicky rozvíjet jeho názorací a představovací schopnosti. V praxi se princip uplatňuje v mnoha formách. Ve vyšších stupních vzdělání se uplatňuje také tzv. nepřímá názornost, vycházející ze zkušeností studentů a také abstraktní názornost například v podobě grafů.

## 2 PEDAGOGICKÉ PRINCIPY

K naplnění tohoto principu má výpočetní technika všechny předpoklady a lepších výsledků může dosahovat jen reálná ukázka jevu nebo skutečného objektu, popřípadě jeho modelu. Způsobů, jak využít možností počítačů je mnoho. Nejjednoduššími přímými metodami může být zobrazení obrázků, přehrání zvuku či video-sekvence. Složitějšími jsou pak různé animace, jejichž parametry lze měnit a ovlivňovat tak tím chování a podmínky virtuálního objektu. I metody využívající nepřímou názornost jsou hojně využívány v počítačových programech. Těmi základními jsou různé grafy. V případě těchto nepřímých metod je však nutné, aby student už měl poměrně dobře vyvinuté kognitivní myšlení.

### 2.5 Uvědomělost

Princip uvědomělosti vytyčuje požadavek, aby student probírané látce a kladeným nárokům plně rozuměl. Zahrnuje v sobě několik podstatných aspektů. Učivo se má ve vědomí studentů odrážet ve formě jasných představ, přesných pojmů, soudů a úsudků. Student má osvojené jevy promýšlet, prostřednictvím myšlenkových operací pochopit jejich podstatu, vztahy mezi jevy a zákonitostmi. Student si má osvojené vědomosti adekvátně vybavovat a správně vyjadřovat přesným užíváním odborného názvosloví, má jasně popsat, definovat a vysvětlit jevy a vyvodit obecné závěry. Také má pochopit úlohu získaných vědomostí a dovedností, správně a tvořivě jich v praxi používat. Smyslem je vyloučit jakékoliv mechanické osvojování.

Tento princip splňuje program v jasném vysvětlení, přesných formulací a náhledů na souvislosti, ale také v přehledném ovládnutí. Samotné ovládací prvky nebo jejich popisky poskytují jasnou představu o svém významu a funkci.

### 2.6 Trvalost

Princip trvalosti požaduje, aby se jednomu osvojené vědomosti, dovednosti, zájmy i postoje staly trvalým majetkem vychovávaného jedince. Nezmiňovanější podmínkou trvalosti je opakování a procvičování. Trvalost však také závisí i na dodržování všech ostatních popsaných principů. Je také podmíněna kvalitou i hloubkou emotivnosti výchovné práce.

Tento princip ovlivňuje tvorbu výukového programu především po pedagogické stránce a jeho celkové kvalitě. Po stránce uživatelského rozhraní je kladen důraz na snadno zapamatovatelné ovládnutí programu, aby student nemusel pokaždé znovu zjišťovat, jak se požadovaná akce provede a mohl se plně věnovat hlavní náplni výukového programu, tedy výukovému procesu.

### 2.7 Přiměřenost

Princip přiměřenosti vyžaduje, aby obsah, formy a metody výchovně vzdělávací práce byly v souladu s věkovou vyspělostí, s dosavadními znalostmi, zkušenostmi a odpovídaly jeho schopnostem. Je třeba znát a respektovat věkové a individuální zvláštnosti.

Východním požadavkem je zachovat základní pravidla pedagogického postupu: od blízkého ke vzdálenějšímu, od známého k neznámému od jednoduchého k složitějšímu

Tato zásada musí přirozeně ovlivňovat tvorbu všech, nejen výukových, programů. Především obsah výukového procesu musí podřízen tomuto principu. Velmi záleží pro koho je daný výukový program určen. Záleží jak na stupni úrovně myšlení daného jednotlivce, což do jisté míry souvisí s věkem. V tomto ohledu je třeba vzít na zřetel i zkušenosti s prací s počítačem.

## 2 PEDAGOGICKÉ PRINCIPY

Přiměřenost tohoto programu se projevila při tvorbě uživatelského rozhraní. Důležité je, aby rozhraní odpovídalo celkové povaze předpokládaného uživatele. V současnosti se velmi často uživatelské rozhraní koncipuje jako hierarchické. Hierarchie znamená, že uživatelé mohou být rozděleni do několika rolí, ve kterých k programu přistupuje. Podle toho jsou přístupné nebo naopak nepřístupné některé funkce programu. Také výukový obsah je zpracován přiměřeně k věku a znalostem studentů 2. ročníku středních škol.

### 2.8 Individuálnost

Tento princip zajišťuje individuální přístup k danému jednotlivci s přihlédnutím k jeho psychickým a sociálním vlastnostem a požadavkům.

Splnění tohoto požadavku vyplývá ze samotného nasazení výpočetní techniky do vzdělávacího procesu. Individuálnost výukového programu spočívá v tempu a v obtížnosti testových úloh. Uživatel může postupovat tímto programem dle vlastního tempa, nebo si může vybrat lehčí, či těžší úroveň závěrečného testu.

### 2.9 Emocionálnost

V poslední době je stále více doceňován pedagogický princip emocionálnosti. Ten vyžaduje probouzet ve výchovně vzdělávacím procesu adekvátní citové požitky studenta. To se může zajistit udržováním příjemné, nekonfliktní a tvůrčí atmosféry během vzdělávacího procesu.

Spojení počítačů a emocionálnosti může vypadat tak trochu nepatříčně, ale je dobré si uvědomit, že to jak celkově program na uživatele působí, může do značné míry ovlivnit celkový výsledek výukového procesu. O tom, jak bude uživatel k práci s daným programem přistupovat rozhoduje především interakce mezi ním a zmiňovaným programem. V případě výukových programů je ovšem také nutnost motivace k absolvování výukového procesu a všechny negativní vlivy mohou neblaze ovlivnit dosažení stanoveného cíle. Moderní pedagogika vyžaduje stále více k aplikaci tohoto principu, který m rozumíme:

- probouzení adekvátních citových prožitků jedince
- udržovat trvale radostnou atmosféru

### 2.10 Jednotnost

Princip jednotnosti výchovného působení informuje o tom, že úspěšná výchova a vzdělávání předpokládá jednotu v požadavcích i v přístupu všech pedagogů, s nimiž se student setkává. Nejednotnost může narušit charakter především u dětí.

Jednost na straně výrobců, tedy koncepcí programu, nelze zaručit. Každý výrobce pracuje mírně odlišnými postupy a má jinou představu o výsledném produktu. Proto lze jednotu zajistit pouze na straně uživatelů vhodnou skladbou výukových produktů, to ovšem vyžaduje často řešení od jednoho výrobce nebo velký výběr podobných produktů, což bývá často problém.

Závěrem lze říci, že význam jednotlivých pedagogických principů pro tvorbu výukových programů je značně rozdílný. Také jejich prosazování a zavádění do praxe se liší u každého z nich.

## 3 Etapizace tvorby výukových programů

### 3.1 Tvorbu a výrobu výukových programů budeme rozdělovat do etap

- Námět
- Projekt
- Příprava obsahu
- Blokové schéma výukového programu
- Scénář a strukturogram výukového dialogu
- Obrazovkový scénář
- Počítačová realizace
- Průběžné a závěrečné hodnocení výukového programu

První čtyři etapy by měly být zpracovány vždy, ať už je to výukový program jakéhokoli typu, třeba i takového, který není popsitelný podrobným scénářem (např. programy simulačního typu)

Námět výukového programu by měl vycházet z projektu výuky a předcházet zpracování projektu programu. Námět by měl obsahovat soubor základních informací, které dovolují výše uvedené otázky posoudit. Nejčastěji bude námět zpracovávat budoucí autor výukového programu, jindy to bude zadavatel. Posuzovateli námětu by měli být odborníci, nezávislí na autorech. Rozhodnutí o doporučení nebo schválení přísluší zaměstnavateli, pokud má být program použit pro vnitřní potřebu, nebo odběrateli, pokud je určen pro vnějšího zákazníka.

#### 3.1.1 Námět by měl obsahovat tyto údaje

- Výstižný název výukového programu
- Cílový počítač (minimální konfigurace, potřebná verze operačního systému)
- Určení (pro který tematický celek, kterého předmětu, kterého ročníku, kterého typu školy)
- Obsah (který hlavní výukový cíl program zajišťuje)
- Typ (které didaktické funkce program plní- např. výklad konzultace, demonstrace počítačového modelu jevu, interaktivní simulace, procvičování, řešení problémů, testování znalostí, didaktická hra)
- Využitelnost (např. frontální výklad učitele, pro kolektivní nebo individuální práce žáků ve třídě nebo v laboratoři, pro samostatnou mimo-rozvrhovou práci žáků ve škole nebo doma)
- Předpokládaná průměrná doba práce s programem
- Krátká anotace výukového programu

#### 3.1.2 Projekt výukového programu

Zatímco účelem přípravy námětu bylo vytvořit podklad pro rozhodnutí, zda má být určitý výukový program vytvořen, projekt je už prvním krokem jeho realizace. Jeho účelem je zformulovat zadání, co má výukový program obsahovat a jaké funkce má zajišťovat. Toto



### 3 ETAPIZACE TVORBY VÝUKOVÝCH PROGRAMŮ

zadání pak bude sloužit autorovi ke shromáždění podkladových materiálů, na něž naváže vlastní tvorba pedagogického scénáře.

Jako záruku dodržení systémového přístupu při projektování výukového programu se doporučuje vyjít z vymezení výukových cílů daného celku.

Na základě analýzy jednotlivých výukových cílů stanovíme ty, při jejichž zajišťování se má počítačový výukový program využít. Ty pak budeme při projektování rozpracovávat. Ke každému z těchto cílů se v projektu uvádí, jaké činnosti s použitím počítače má student provést, aby daného výukového cíle dosáhl, a jaké didaktické funkce přitom počítač zajišťuje.

Výukové cíle mají být vymezovány hierarchizovaně, chceme-li dodržet systémový přístup k výuce.

Ve výukových programech je třeba omezovat ty části, které zajišťují didaktické funkce prezentace látky a větší pozornost věnovat didaktickým funkcím procvičování, řízení i samostatnému řešení úloh, kontrole dosažení výukových cílů, diagnostice chybných výkonů a korekci.

#### **3.1.3 Příprava obsahu výukového programu**

V navazující etapě musí autor pro tyto činnosti a funkce připravit konkrétní náplně. Stejně jako v případě projektu i ve fázi přípravy obsahu se doporučuje shromažďovat materiál k jednotlivým výukovým cílům. Základním rysem etapy přípravy obsahu je jednodušších typů výukových programů sbírání textů a obrázků. Část materiálu se dá získat z již dostupných zdrojů, část podkladů si autoři musí vytvořit speciálně pro účely výukového programu.

#### **3.1.4 Blokové schéma výukového programu**

Úkolem této etapy je návrh vnitřní struktury na úrovni modulů. Na této úrovni může být vnitřní struktura výukového programu rozčleněna buď primárně podle výukových cílů, nebo primárně podle didaktických funkcí, nebo může být použito struktury smíšené.

#### **3.1.5 Ověření v pedagogické praxi**

V návaznosti na vytvoření didaktických metodických materiálů se předpokládá jejich následné ověření v pedagogické praxi. Uskuteční se za účasti vyučujících určených studijních předmětů v průběhu ucelených etap výuky s následným vyhodnocením účinnosti. Výsledky tohoto ověření budou základem pro doporučení, resp. jako východiska pro další, navazující etapy řešení tématu a propracování metodických opor.

## 4 Požadavky

### 4.1 Hardwarové požadavky

Na stanici jsou shodné s požadavky na běžný kancelářský stroj s operačním systémem Windows 2000/XP.

### 4.2 Softwarové požadavky

#### 4.2.1 Vývojové požadavky

Na PC musí být nainstalovaný .NET Framework 3.5 SP1 a Visual Studio .NET 2008 SP1 spolu s Microsoft Chart Controllerem. Pro tvorbu HTML stránek stačí poznámkový blok, ale raději použijte PSPad.

#### 4.2.2 Provozní požadavky

V PC mít nainstalovaný .NET Framework 3.5 SP1 a Microsoft Chart Control.

### 4.3 Požadavky na webovou podporu

Webové stránky by měly být funkční, měly by se zde vyskytovat odkazy na rozšiřující informace. Web by měl být přehledný a srozumitelný. Jako hlavní stránka bude stručný popis programu a nějaké náhledy jak program vypadá. Bude zde uveřejněna nejaktuálnější verze programu a program zde bude ke stažení. Součástí webové podpory bude i nápověda k programu, kde bude vysvětlena veškerá manipulace s programem jednoduchými kroky. Poslední částí budou kontaktní informace, kde budou moci návštěvníci zasílat dotazy, náměty a jiné připomínky k programu.

### 4.4 Požadavky na uživatele

Uživatel by měl umět základní operace s počítačem a základní znalost problematiky funkcí, aby mohl vůbec aplikaci používat a věděl kde a co zadat. Toto ale bude také popsáno v nápovědě.

### 4.5 Funkční požadavky

- zadat vlastní funkci
- výuka teorie níže uvedených vlastností a typů funkcí
- vypsání níže uvedených vlastností zadané funkce
- vykreslení zadané funkce do grafu
- testování vědomostí formou testu
- správa testových otázek
- správa uživatelů

#### 4.5.1 Vlastnosti funkcí

- rostoucí
- klesající
- konstantní
- sudá
- lichá
- periodická

## 4 POŽADAVKY

- prostá (monotónní)

### 4.5.2 Typy funkcí

- Lineární
- Kvadratická
- Mocninná
- Logaritmická
- Exponenciální
- Goniometrická

### 4.6 Programátorské minimum

V programu musí rozhodně být vykreslení zadané funkce do grafu a jejích vlastností. Ve výukové oblasti musí být určité vysvětlena daná problematika minimálně pomocí textu a názorných příkladů. Testovací část bude formou testu, kde bude položena otázka, a bude na výběr 5 možností, kdy jedna odpověď z těchto 5ti možností bude ta správná. Program bude určitě minimálně v českém jazyce. Webové stránky budou obsahovat seznámení s programem. Stručnější vysvětlení problematiky vyučované v programu. Stránky budou v českém jazyce a kontakt, kde se budou moci uživatelé zeptat na dotazy či poradit co by chtěli v dalších verzích programu.

### 4.7 Programátorské maximum

V Programu bude možnost volby anglického jazyka. Program by vypočítal i vlastnosti, které se na středních školách většinou nevyučují a studenti se s nimi seznámí až na vysokoškolském studiu jako např. asymptoty funkcí, 1. i 2. Derivací, výpočet limit v +/- nekonečno, nalezení nulových bodů. V testovací části by se mohli vyskytovat i otázky kde by se zadávaly výsledky a program by je pak opravil.

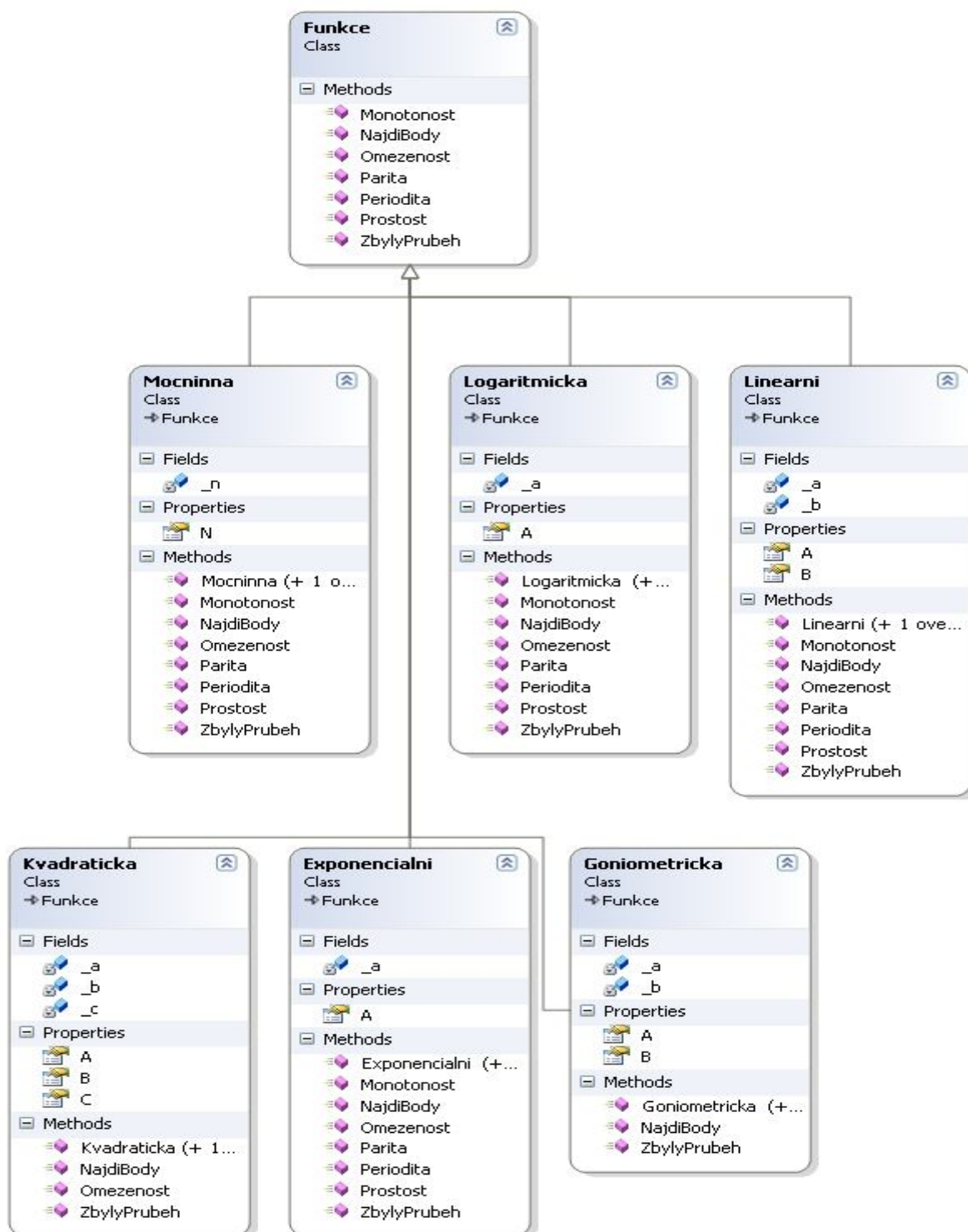
### 4.8 Nefunkční požadavky

Program bude naprogramován v jazyce C# za pomoci .NET Framework 3.5 SP1 ve vývojovém prostředí Visual Studio 2008. Webové stránky k tomuto programu budou vytvořené pomocí HTML jazyku a CSS stylů. Základní verze by měla být napsána hlavně v českém jazyce. Dalším jazykem by mohla být angličtina, ale to není prioritou.

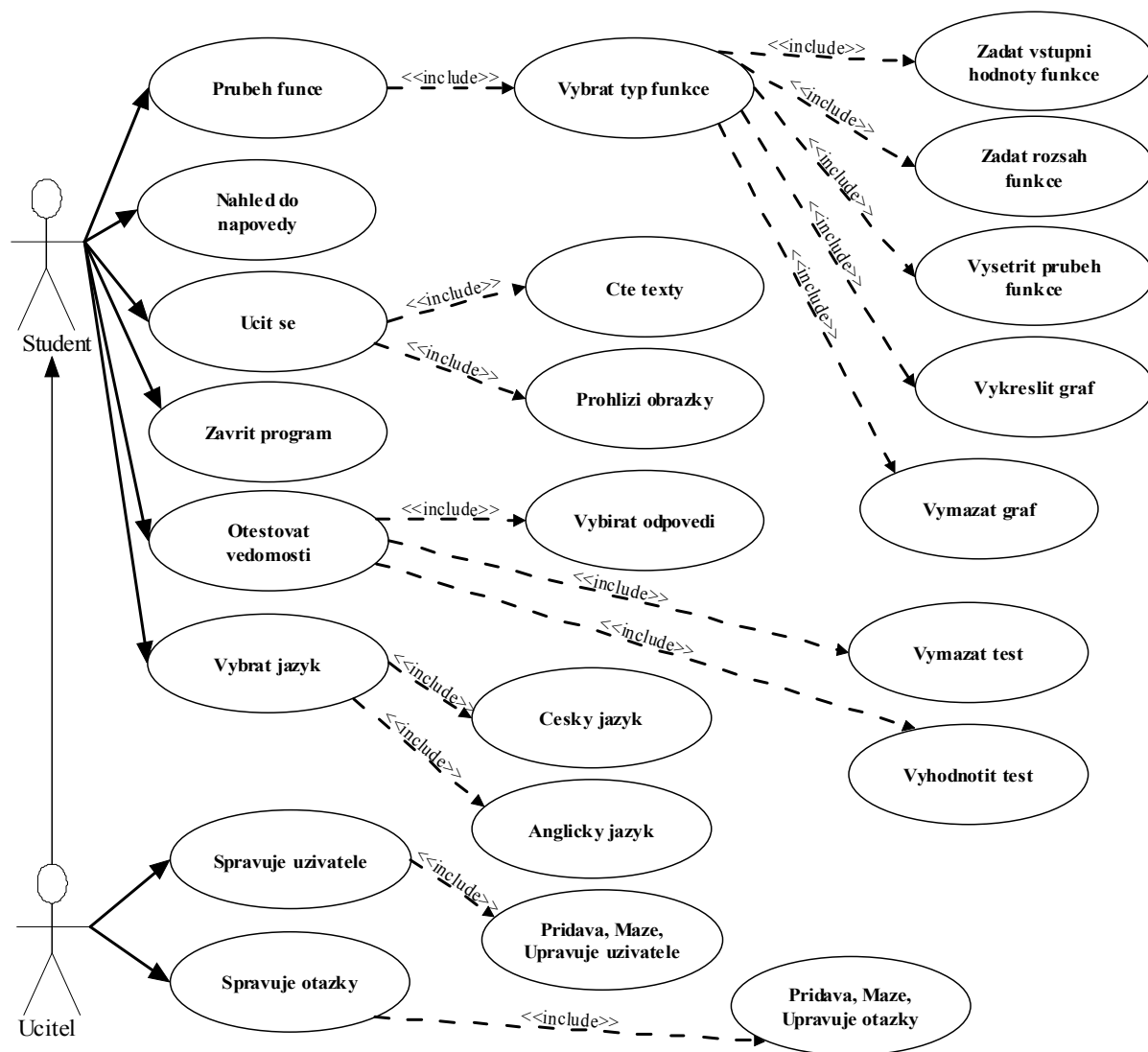
# 5 Analýza

## 5.1 Diagramy

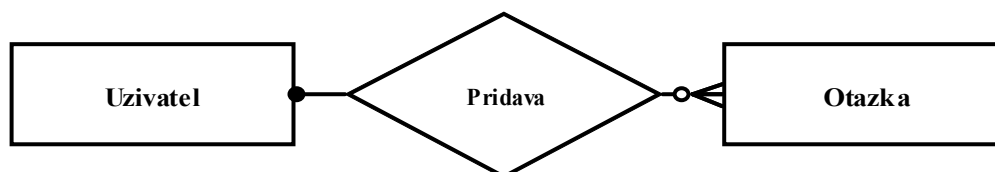
### 5.1.1 Třídní diagram



## 5.1.2 Use-case diagram



## 5.1.3 ER Diagram



## 5.1.4 Lineární zápis

- primární klíč, *cizí klíč*

Otazka (id\_otazky, id\_uzivatele, zadani, odpoved\_a, odpoved\_b, odpoved\_c, odpoved\_d, odpoved\_e, spravna\_odpoved, typ\_funkce)

Uzivatel (id\_uzivatele, login, heslo, role)

## 5.1.5 Lineární zápis typů vztahů

PRIDAVA(Uzivatel, Otazka) 1:N

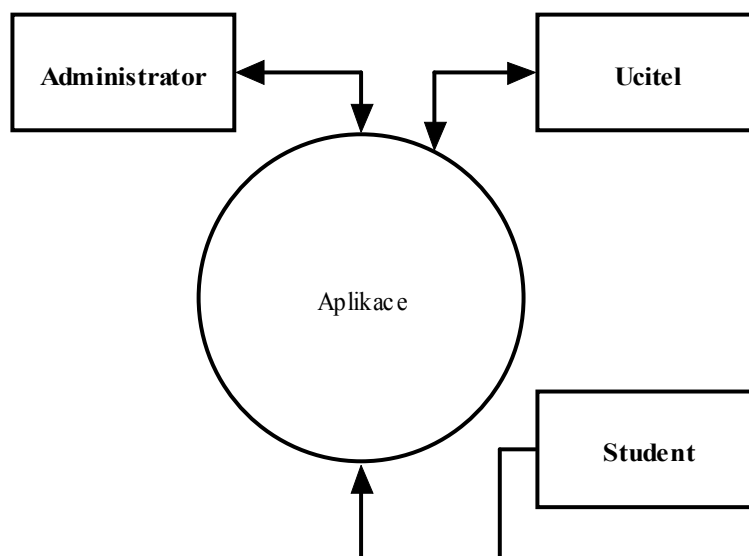
## 5.2 Datová analýza (datový slovník)

Otazka							
Název	Typ	Velikost	Klíč	Null	Index	IO	Popis
id_otazky	int		ANO	NE	ANO		identifikátor otázky
id_uzivatele	int		NE	NE	NE		cizí klíč z tabulky uzivatel
zadani	varchar	500	NE	NE	NE		
odpoved_a	varchar	500	NE	NE	NE		
odpoved_b	varchar	500	NE	NE	NE		
odpoved_c	varchar	500	NE	NE	NE		
odpoved_d	varchar	500	NE	NE	NE		
odpoved_e	varchar	500	NE	NE	NE		
spravna_odpoved	varchar	500	NE	NE	NE		
typ_funkce	varchar	50	NE	NE	NE		

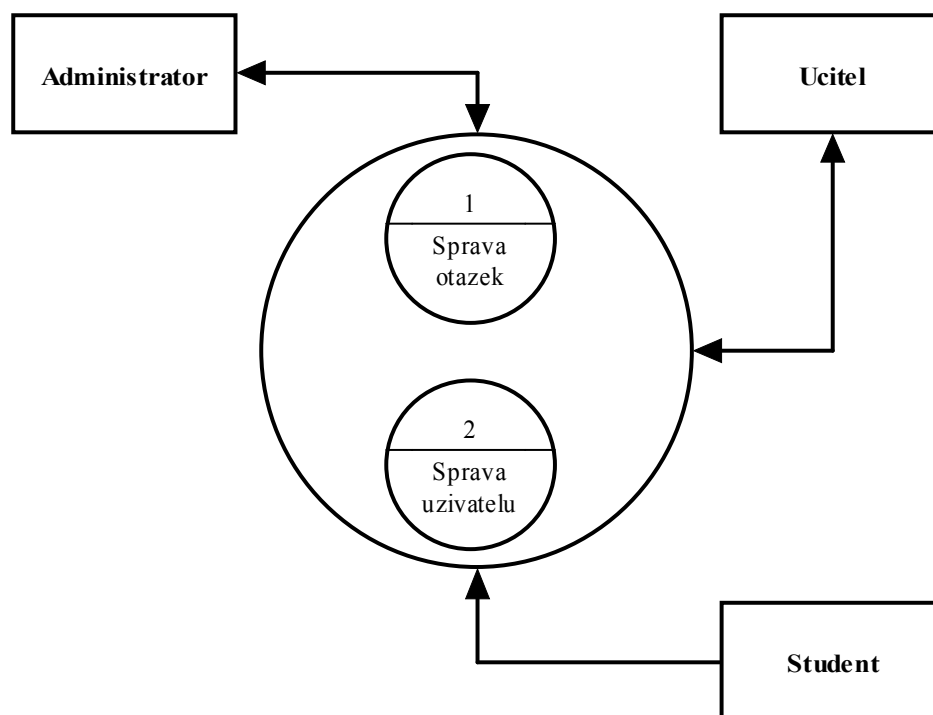
Uzivatel							
Název	Typ	Velikost	Klíč	Null	Index	IO	Popis
id_uzivatele	int		ANO	NE	ANO		identifikátor uživatele
login	varchar	50	NE	NE	NE		
heslo	varchar	50	NE	NE	NE		
role	int		NE	NE	NE	0,1,2	0-admin, 1-ucitel, 2-student

## 5.3 Funkční analýza

### 5.3.1 Kontextový diagram

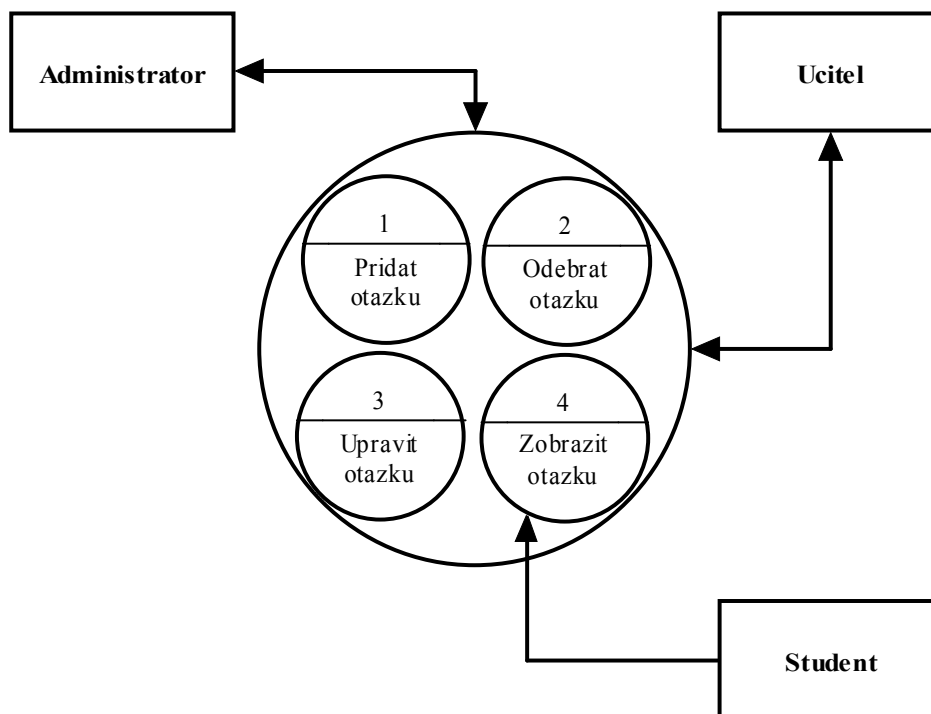


5.3.2 0. Úroveň

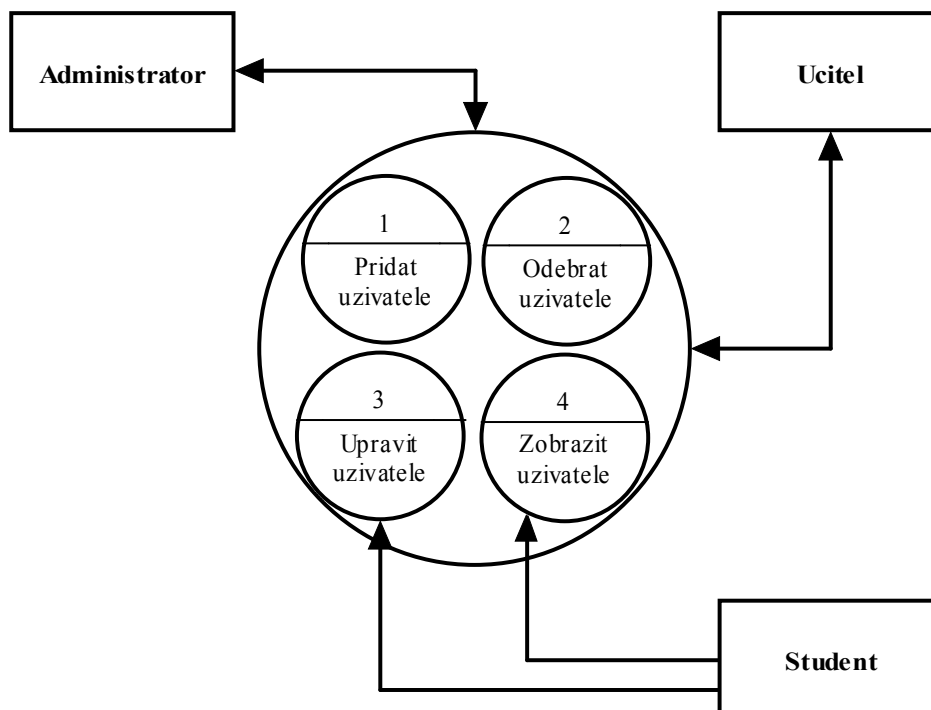


### 5.3.3 1. Úroveň

- Správa otázek



- Správa uživatelů





# 6 Implementace

## 6.1 Technologie

Systém je implementovaný v prostředí .NET s využitím jazyka C# a framework verze 3.5 SP1. Program je tvořen klasickými WinForm (windows formuláři), kde používám standardní ovládací prvky. Webová podpora je vytvořena v HTML jazyce a za použití CSS stylů je oddělena designová stránka od obvyčejného textu.

### 6.1.1 .NET Framework 3.5 SP1

Balíček Microsoft .NET Framework obsahuje vše potřebné ke spuštění aplikací vyvinutých pro prostředí Microsoft .NET. Takové programy totiž ke svému spuštění potřebují mít nainstalovány knihovny tohoto prostředí, a to právě zaručí tento balík. Verze Microsoft .NET Framework 3.5, která byla dána do oběhu v únoru 2008 je rozšířením verze 2.0 a to velmi zásadním.

Program je tedy vyvíjen v tomto pracovním rámci. Projekt je rozdělen na dvě části, v první jsou formuláře prezentační vrstvy a ve druhé jsou logické třídy programu, ve kterých se pracuje se vstupními daty. Logická vrstva vytvoří soubor DLL, který nareferencujeme do hlavní prezentační vrstvy. DLL soubor se skládá z 6ti tříd (každá pro jednotlivý typ funkce), které dědí funkce z hlavní třídy. V hlavní třídě jsou použity virtuální funkce a pokud dědicí třídy budou chtít samy implementovat funkce, které se jmenují stejně jako funkce z třídy hlavní musí použít klíčového slova override.

### 6.1.2 C#

Vysoce úrovněový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework, později schválený standardizačními komisemi ECMA a ISO. Microsoft založil C# na jazycích C++ a Java (a je tedy nepřímým potomkem jazyka C, ze kterého čerpá syntaxi).

### 6.1.3 HTML

HyperText Markup Language, označovaný zkratkou HTML, je značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu. Jazyk je aplikací dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

Webová podpora je tedy vytvořena v tomto značkovacím jazyce a skládá se z 4 html stránek a 1 css souboru. Struktura webu je takováto:

- index.html
- download.html
- help.html
- kontakt.html
- style.css
- /img                    - složka s obrázky

### 6.1.4 CSS

Je zkratka pro anglický název Cascading Style Sheets, česky tabulky kaskádových stylů. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML. Jazyk byl navržen standardizační organizací W3C, autorem prvotního návrhu byl Håkon Wium Lie. Byly vydány zatím dvě úrovně specifikace CSS1 a CSS2, dokončuje se revize CSS 2.1 a pracuje se na verzi CSS3. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí.

### 6.1.5 SQL

SQL (někdy vyslovováno anglicky es-kjů-el) je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. SQL je zkratka anglických slov Structured Query Language (strukturovaný dotazovací jazyk).

Jelikož se v testovací části nahrávají otázky z databáze, je třeba použít SQL jazyk pro nahrání dat. Databázi používám i co se týče správy uživatelů a správy testových otázek. Práce s databázemi v prostředí MSSQL je poměrně jednoduchá a přehledná.

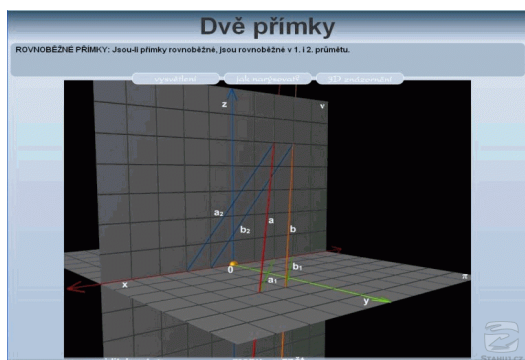
### 6.1.6 XML

XML (eXtensible Markup Language, česky rozšiřitelný značkovací jazyk) je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a široké spektrum různých typů dat. Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Jazyk umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, nezabývá se sám o sobě vzhledem dokumentu nebo jeho částí. Prezentace dokumentu (vzhled) se potom definuje pomocí kaskádových stylů. Další možností je pomocí různých stylů provést transformaci do jiného typu dokumentu, nebo do jiné struktury XML.

V mém programu jsem s XML pracoval v souvislosti s vícejazyčnou podporou. Veškeré texty v programu jsou uloženy ve strukturovaném xml souboru. Pokud bychom chtěli časem přidat další podporu pro další cizí jazyk, není třeba v programovém kódu měnit všude všechny texty, ale jednoduše se přeloží pouze xml soubor.

## 6.2 Inspirace z jiných programů

### 6.2.1 Deskriptivní geometrie 2.02

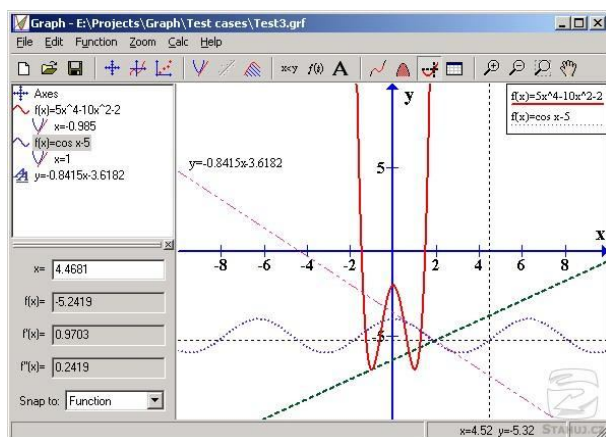


Tento program, jak už název napovídá, se zabývá výukou deskriptivní geometrie. Jedná se o program s minimálními nároky na výkon počítače, který využívá nejmodernější multimediální technologie. Každá úloha je vymodelována pomocí 3D grafiky s možností sledovat úlohu z různých úhlů. Také jsou zde animace, které vysvětlují jak úlohu řešit v sešitě. V programu je zpracováno učivo přibližně jednoho roku výuky deskriptivní geometrie na

## 6 IMPLEMENTACE

střední škole. Velkou výhodou tohoto programu je použití animací při vysvětlování dané problematiky. Slabší stránkou programu je ale slabá možnost otestování nabytých vědomostí. Otázky v testu jsou stále se opakující dokola což je možná způsobeno tím, že mám pouze standart verzi. Každopádně v mém programu bych chtěl rozhodně více otestovat znalosti uživatele.

### 6.2.2 Graph

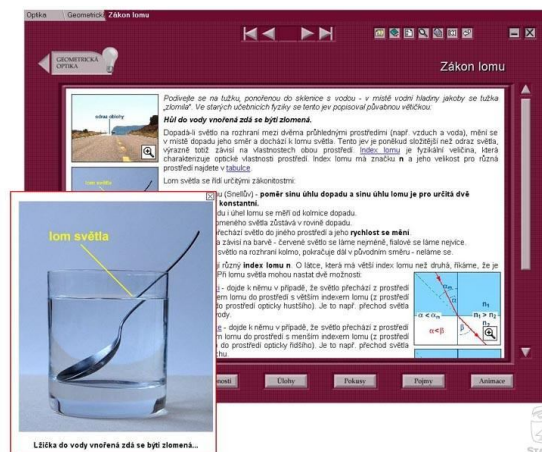


Program pro kreslení grafů a počítání délek křivek, obsahů oblastí pod křivkami, derivací funkcí, doplňování regresních křivek k posloupnostem bodů apod. Tento program je pro mě inspirací ve směru vykreslování grafů. To je taky vlastně jediná funkce tohoto programu protože se nejedná o program výukový. Program vykreslí na plátně graf podle zadané funkce a můžeme zde i přidávat další sečny, tečny apod. Jelikož se ale nejedná o program výukový tak postrádá

teoretickou část a část kde se dá teorie otestovat.

### 6.2.3 Fyzika zajímavě – Optika

Program nemá nahrazovat učebnici se spoustou vzorečků a výpočtů, jeho cíle jsou spíše motivační. Obsahuje asi 50 článků na úrovni základní školy, rozdělených do 10 kapitol. Pro středoškoláky a vážnější zájemce o fyziku je zde oddíl „Teorie“. K oživení učiva uvádí autor v sekci „Zajímavosti“ asi 100 článků, obsahujících různé aplikace i kuriozity ve vědě, technice či v domácnosti. Sekce „Pokusy“ obsahuje náměty na jednoduché experimenty. Se životy významných fyziků seznamuje oddíl „Osobnosti“. Významnou částí programu jsou animace, které názorně předvádějí optické jevy. Na programu se mi líbí v zásadě asi vše, protože má všechno co má pravý výukový program mít. Jediný důvod proč bych ho tu ale neměl představovat je, že se jedná o program na výuku fyziky. Z pohledu zaměřeného na výuku je ale program asi nejbližší tomu jak by měl vypadat program můj.



## 6.3 Předvedení programu

### 6.3.1 Přihlašovací vstupní formulář

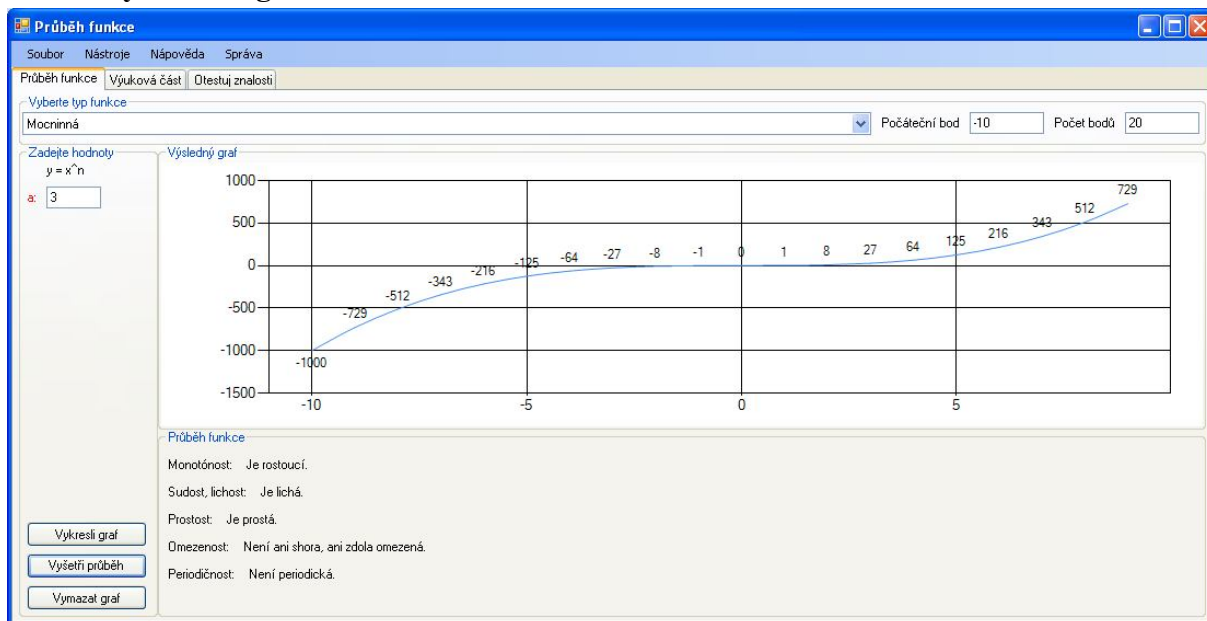
Tento formulář se zobrazí, pokud se budete chtít dostat ke správě uživatelů nebo správě testových otázek. Formulář tvoří dvě tlačítka. Pro přihlášení je tlačítko „Log In“, které zkontroluje zadané údaje, jestli se shodují s údaji v databázové tabulce Uživatelé a v případě

## 6 IMPLEMENTACE

nalezení shody se okno zavře a otevře se správa bud' samotných uživatelů, nebo správa otázek. Tlačítko „Cancel“ jak už název napovídá, zavře přihlašovací okno bez provedení jakékoliv akce. Tímto formulářem vlastně zabráníme tomu aby uživatel, který program neinstaloval se nedostal ke správě a nemohl tak podniknout záškodnické akce. V moderních aplikacích je již víceuživatelský přístup hojně využíván.

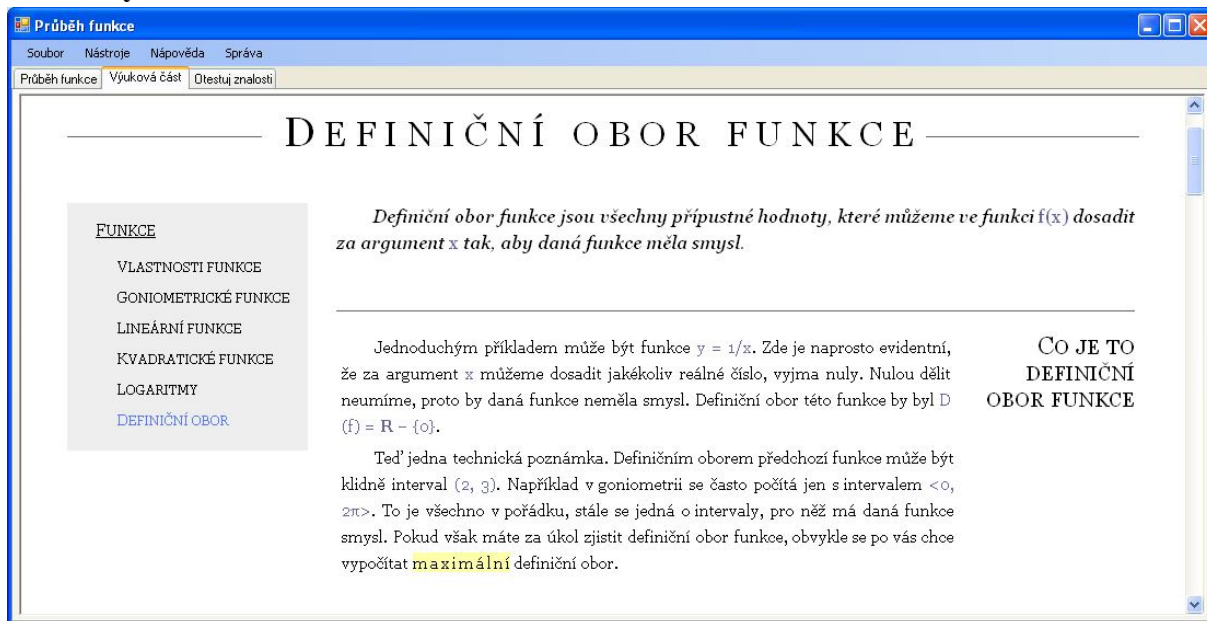


### 6.3.2 Vykreslení grafu



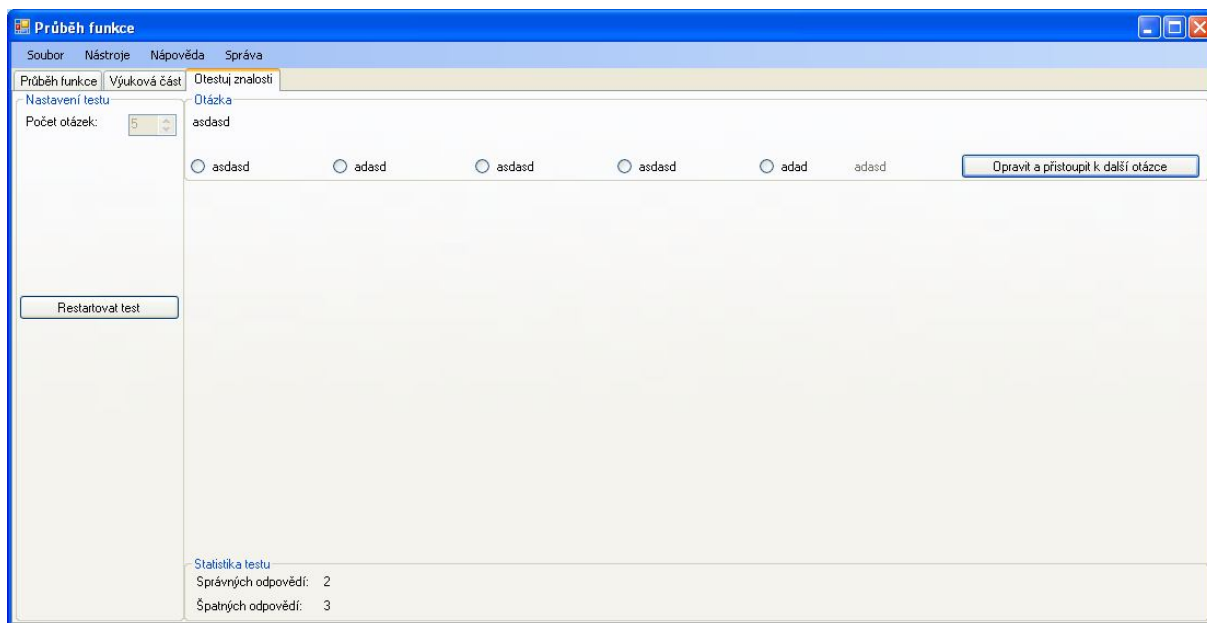
Zde je vidět názorná část programu, kde si uživatel může zadávat vstupní parametry a po stisknutí tlačítka „Vykreslit graf“ se mu funkce podle zadaných parametrů vykreslí do grafu. Dále má uživatel možnost vyšetřit průběh funkce pomocí tlačítka „Vyšetřit průběh“, kdy se uživatel dozví základní vlastnosti funkce. Mezi tlačítky nesmí chybět možnost vymazání grafu, která se provede klepnutím na tlačítko „Vymazat graf“.

### 6.3.3 Výuková část



Na tomto obrázku si můžeme prohlédnout výukovou část programu, která se skládá z webového prohlížeče, který zobrazuje webové html stránky s obrázky a názornými příklady probírané látky v programu.

### 6.3.4 Testovací část



Testovací část programu je tvořena třemi částmi. V první části si uživatel nastaví potřebné parametry testu jako je počet otázek, které se mu do testu vygenerují. Dále zde má tlačítko pro spuštění samotného testu, po jehož kliknutí se zpřístupní druhá část výukového okna, kterým je vygenerování samotné otázky. V této části se nám zobrazí zadání vygenerované otázky a pod zadáním vidíme 5 možností správné odpovědi, kdy vždy jedna je správná. Pokud budeme chtít označit správnou odpověď, budeme muset zakliknout jeden z těchto pěti radio buttonů. Oprava otázky probíhá ihned po kliknutí na tlačítko „Opravit a přistoupit k další otázce“. Jakmile dosáhne počet zobrazených otázek počtu který jsme nastavili na začátku,

zobrazí se informační okno s oznámením že jsme dokončili test a zobrazí se nám poslední část kterou je statistika testu s vyobrazením počtu správných a špatných odpovědí.

### 6.3.5 Správa otázek

Ve správě otázek máme tlačítka pro základní funkce správy těchto otázek. Tlačítka vyobrazenými jako šipky doleva a doprava se pohybujeme podle indexu otázek v databázi. V průběhu procházení tabulky otázek se nám pro každou otázku zobrazí její atributy a data uložená v tabulce a máme tedy možnost uložená data libovolně upravovat, kromě atributu `id_otázky` jelikož s tím by uživatel neměl hýbat a nechat ho tak jak je. Právě proto je i vyobrazen pouze jako label a není k němu přístup. Dalšími tlačítky jsou přidání zobrazené jako znaménko „+“, které přidá do tabulky otázek nový záznam, a do atributů otázky budeme moci pomocí prázdných textových polí zadat vstupní hodnoty. Tlačítko vyobrazené jako červené X odstraní právě prohlížený záznam.

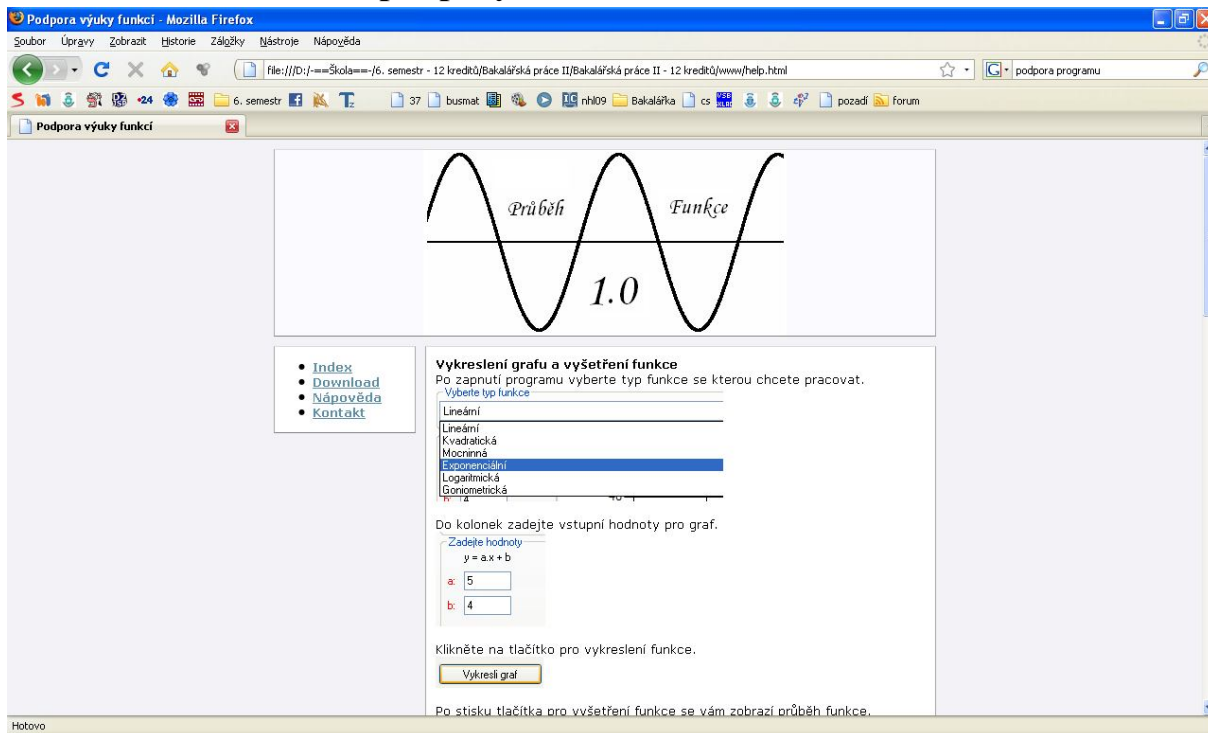
### 6.3.6 Správa uživatelů

Ve správě uživatelů máme tlačítka pro základní funkce správy těchto uživatelů. Tlačítka vyobrazenými jako šipky doleva a doprava se pohybujeme podle indexu uživatelů v databázi. V průběhu procházení tabulky uživatelů se nám pro každého uživatele zobrazí jeho atributy a data uložená v tabulce a máme tedy možnost uložená data libovolně upravovat, kromě atributu `id_uzivatele` jelikož s tím by uživatel neměl hýbat a nechat ho tak jak je. Právě proto je i



vyobrazen pouze jako label a není k němu přístup. Dalšími tlačítka jsou přidání zobrazené jako znaménko „+“, které přidá do tabulky otázek nový záznam, a do atributů uživatele budeme moci pomocí prázdných textových polí zadat vstupní hodnoty. Tlačítko vyobrazené jako červené X odstraní právě prohlížený záznam. Můžeme si tedy všimnout, že práce se správou uživatelů je stejná jako se správou otázek což ulehčuje uživateli zvyknout si na způsob této práce a když se naučí pracovat se správou jedné části, automaticky umí spravovat i části ostatní.

## 6.4 Předvedení webové podpory softwaru



Na tomto obrázku si pouze ukážeme, jak vypadá výsledná webová podpora pro tento program. Webová stránka je rozložena do částí hlavička, menu, obsah, patička. V hlavičce máme logo tohoto programu, v menu jsou 4 odkazy na další stránky v rámci webu, v obsahu se vždy zobrazuje obsah stránky a v patičce jsou uvedeny jen údaje a informace o vlastníkovu webu.

## 7 Závěr

V této bakalářské práci jsme viděli postup při tvorbě výukového softwaru a jeho vývoj na platformě .NET za použití jazyka C#. Jako výsledek této práce byl vytvořen program rozdělený do 3 částí, názorná, ve které je si student za pomoci vykreslení grafu a následného vyšetření zadané funkce může prohlédnout, jak daná funkce vypadá, výuková, kde student za pomoci textů a obrázků se dozví o dané problematice a testovací, kde si může v podobě vygenerovaných otázek vyzkoušet své nabitě vědomosti. Dále byla vytvořena webová podpora výukového programu, kde se nachází program ke stažení a vyzkoušení. Na webové stránce se nachází i podrobná nápověda k užívání programu. V tomto programu se mi tedy podařilo vytvořit výukový software, který splňuje aspoň minimální požadavky, které jsem si na začátku nastavil. V průběhu tvorby softwaru jsem přidal i některé, se kterými jsem nepočítal, ale později mi přišlo za vhodné je do programu přidat.



## 8 Literatura

- [1] John Sharp, *Microsoft Visual C# 2005 Krok za krokem*, 2006
- [2] Doc. RNDr. Oldřich Odvárko, DrSc., *Matematika pro Gymnázia Funkce*, 2000
- [3] Jana Šarmanová, *Databázové informační systémy učební text*, Ostrava 2007
- [4] Vondrák, Ivo: *Úvod do softwarového inženýrství*, Ostrava, 2002
- [5] Vondrák, Ivo: *Metody byznys modelování*, Ostrava 2004
- [6] Výukové programy o elektrotechnice vhodné k aplikaci v technické výchově  
<<http://www.elektrotechnicestavebnice.xf.cz/programy.htm>>
- [7] Jiří Kofránek, Michal Andrlík, Tomáš Kripner, Zdeněk Wunsch, Tvorba výukových multimediálních programů  
<[http://patf-biokyb.lf1.cuni.cz/publications/tvorba\\_multimed\\_vyuk\\_prg.pdf](http://patf-biokyb.lf1.cuni.cz/publications/tvorba_multimed_vyuk_prg.pdf)>
- [8] Jiří Dostál, Výukový software a počítačové hry – Nástroje moderního vzdělávání  
<[http://www.jtie.upol.cz/clanky\\_1\\_2009/dostal.pdf](http://www.jtie.upol.cz/clanky_1_2009/dostal.pdf)>
- [9] Gejzda Dohnal, Výukové programy na INTERNETu  
<<http://www.statapol.cz/stakan/stakan-11-1999/dohnals2.pdf>>
- [10] Bořivoj Brdička, Učení s počítačem  
<<http://it.pedf.cuni.cz/~bobr/ucspoc/>>
- [11] Výukový program  
<[http://cs.wikipedia.org/wiki/V%C3%BDukov%C3%BD\\_program](http://cs.wikipedia.org/wiki/V%C3%BDukov%C3%BD_program)>